

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

IN RE APPLICATION OF: Yoshikazu OGAWA, et al.

GAU:

SERIAL NO: New Application

EXAMINER:

FILED: Herewith

FOR: PROCESSOR HAVING REGISTER RENAMING FUNCTION

REQUEST FOR PRIORITY

COMMISSIONER FOR PATENTS
ALEXANDRIA, VIRGINIA 22313

SIR:

- ☐ Full benefit of the filing date of U.S. Application Serial Number _____, filed _____, is claimed pursuant to the provisions of 35 U.S.C. §120.
- ☐ Full benefit of the filing date(s) of U.S. Provisional Application(s) is claimed pursuant to the provisions of 35 U.S.C. §119(e):
Application No. _____ Date Filed _____

☒ Applicants claim any right to priority from any earlier filed applications to which they may be entitled pursuant to the provisions of 35 U.S.C. §119, as noted below.

In the matter of the above-identified application for patent, notice is hereby given that the applicants claim as priority:

<u>COUNTRY</u>	<u>APPLICATION NUMBER</u>	<u>MONTH/DAY/YEAR</u>
Japan	2003-096751	March 31, 2003

Certified copies of the corresponding Convention Application(s)

- ☒ are submitted herewith
- ☐ will be submitted prior to payment of the Final Fee
- ☐ were filed in prior application Serial No. _____ filed _____
- ☐ were submitted to the International Bureau in PCT Application Number _____
Receipt of the certified copies by the International Bureau in a timely manner under PCT Rule 17.1(a) has been acknowledged as evidenced by the attached PCT/IB/304.
- ☐ (A) Application Serial No.(s) were filed in prior application Serial No. _____ filed _____; and
- ☐ (B) Application Serial No.(s)
☐ are submitted herewith
☐ will be submitted prior to payment of the Final Fee

Respectfully Submitted,

OBLON, SPIVAK, McCLELLAND,
MAIER & NEUSTADT, P.C.



Marvin J. Spivak

Registration No. 24,913

C. Irvin McClelland
Registration Number 21,124

Customer Number

22850

Tel. (703) 413-3000
Fax. (703) 413-2220
(OSMMN 05/03)

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日
Date of Application:

2003年 3月31日

出 願 番 号
Application Number:

特願2003-096751

[ST.10/C]:

[JP2003-096751]

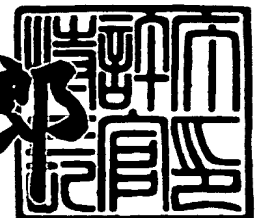
出 願 人
Applicant(s):

東芝マイクロエレクトロニクス株式会社
株式会社東芝

2003年 4月18日

特 許 庁 長 官
Commissioner,
Japan Patent Office

太田信一郎



出証番号 出証特2003-3028163

【書類名】 特許願

【整理番号】 14157001

【提出日】 平成15年 3月31日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 9/46

【発明の名称】 レジスタ・リネーミング機能を有するプロセッサ

【請求項の数】 6

【発明者】

 【住所又は居所】 神奈川県川崎市川崎区駅前本町 2 5 番地 1 東芝マイクロ
 ロエレクトロニクス株式会社内

 【氏名】 小 川 義 和

【発明者】

 【住所又は居所】 神奈川県川崎市幸区小向東芝町 1 番地 株式会社東芝
 マイクロエレクトロニクスセンター内

 【氏名】 後 藤 治 隆

【特許出願人】

 【識別番号】 000221199

 【住所又は居所】 神奈川県川崎市川崎区駅前本町 2 5 番地 1

 【氏名又は名称】 東芝マイクロエレクトロニクス株式会社

【特許出願人】

 【識別番号】 000003078

 【住所又は居所】 東京都港区芝浦一丁目 1 番 1 号

 【氏名又は名称】 株式会社 東 芝

【代理人】

 【識別番号】 100075812

 【弁理士】

 【氏名又は名称】 吉 武 賢 次

【選任した代理人】

 【識別番号】 100088889

【弁理士】

【氏名又は名称】 橘 谷 英 俊

【選任した代理人】

【識別番号】 100082991

【弁理士】

【氏名又は名称】 佐 藤 泰 和

【選任した代理人】

【識別番号】 100096921

【弁理士】

【氏名又は名称】 吉 元 弘

【選任した代理人】

【識別番号】 100103263

【弁理士】

【氏名又は名称】 川 崎 康

【手数料の表示】

【予納台帳番号】 087654

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 レジスタ・リネーミング機能を有するプロセッサ

【特許請求の範囲】

【請求項 1】

命令を取り込む命令フェッチ部と、前記命令フェッチ部により取り込まれた前記命令から命令コードを復号するデコード部と、前記デコード部により復号された前記命令コードにより示されるレジスタ番号に対応するデータを保持するレジスタ部とを備え、デコードされた命令コードに含まれる論理レジスタ番号とその番号の物理レジスタ番号との対応をレジスタエイリアステーブルに保持すると共に前記物理レジスタ番号の割り当て可能な番号を物理レジスタフリーリストに格納して命令番号、処理状態、アーキテクチャレジスタ番号、古い物理レジスタ番号の対応関係を格納するリオーダーバッファと前記物理レジスタフリーリストとを用いてレジスタエイリアステーブルの内容を順次書き換えるレジスタ・リネーミング機能を有するプロセッサにおいて、

前記命令コードにより示されるレジスタ番号に対応するデータを保持するレジスタ本体と、

前記レジスタ本体に保持された内容の一部をキャッシングするキャッシングレジスタと、

前記命令フェッチ部からの命令により前記キャッシングレジスタにより保持されている論理レジスタ番号やキャッシングレジスタ番号を含む内部命令の状態に関する情報を保持する内部命令情報保持部と、

前記命令フェッチ部からの命令をプリデコードした命令コードと前記内部命令情報保持部により保持された前記内部命令の状態に関する情報とを比較して内部命令を挿入するか否かを判定する命令挿入判定部と、

前記命令挿入判定部により内部転送命令を挿入すべきものと判定されたときに前記キャッシングレジスタと前記レジスタ本体との間で内部データの転送を行なうレジスタ転送命令を発行するレジスタ転送命令発行部と、

を備えることを特徴とするレジスタ・リネーミング機能を有するプロセッサ。

【請求項 2】

前記レジスタ本体はプログラム上で参照可能な論理レジスタより構成され、前記キャッシングレジスタは前記論理レジスタの一部を保持する内部レジスタより構成され、前記レジスタ転送命令発行部は前記論理レジスタと前記内部レジスタとにより使用される内部レジスタ番号に変換する変換部と前記論理レジスタと前記内部レジスタとの間でデータ転送を行なうプロセッサ内部命令コードと同じ形式のコードを生成するコード生成部とより構成されることを特徴とする請求項 1 に記載のレジスタ・リネーミング機能を有するプロセッサ。

【請求項 3】

前記命令フェッチ部からの命令を命令コードにプリデコードするプリデコード部をさらに備えることを特徴とする請求項 1 に記載のレジスタ・リネーミング機能を有するプロセッサ。

【請求項 4】

前記内部命令情報保持部、前記プリデコード部、前記命令挿入判定部、前記レジスタ転送命令発行部によりロードレジスタ命令とストアレジスタ命令を挿入して命令を発行するレジスタ命令挿入ユニットが構成されていることを特徴とする請求項 3 に記載のレジスタ・リネーミング機能を有するプロセッサ。

【請求項 5】

前記レジスタ命令挿入ユニットは、前記内部命令情報を保持する内部命令保持部と、前記命令フェッチ部より供給される命令コードからレジスタ番号情報を取り出すプリデコード部と、このプリデコード部からの前記レジスタ番号情報と前記内部命令保持部が保持する内部命令情報の T A G 領域に格納される論理レジスタ番号の比較を行なって挿入命令のレジスタ番号を生成する前記命令挿入判定部としての挿入命令レジスタ番号生成部と、この挿入命令レジスタ番号生成部により生成された前記挿入命令のレジスタ番号に基づいてロード／ストアレジスタ命令を発行する前記レジスタ転送命令発行部としてのロード／ストアレジスタ命令発行部と、を備えることを特徴とする請求項 4 に記載のレジスタ・リネーミング機能を有するプロセッサ。

【請求項 6】

前記命令フェッチ部と前記命令デコード部との間に、前記ロード／ストアレジ

スタ命令発行部により発行された命令に基づいて、前記命令フェッチ部より供給される命令の前に前記ロード／ストアレジスタ命令発行部からの命令を付加して前記命令デコード部に供給する命令挿入制御部をさらに備える請求項5に記載のレジスタ・リネーミング機能を有するプロセッサ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明はレジスタ・リネーミング機能を有するマイクロプロセッサに関する。

【0002】

【従来の技術】

一般に、マイクロプロセッサにおいては、アウト・オブ・オーダー（out-of-order—順番通りでない—）実行や、レジスタ・リネーミング（register renaming—レジスタ名の変更—）機能等の複雑な機能を組み込むことによってプロセッサの性能向上を図っている。レジスタ・リネーミング機能は、レジスタの複製を作成しておくことにより、WAR, WAWといったレジスタ書き込みにより引き起こされるハザード（hazard—障害・事故—）を解消するためプログラム上では参照することができない物理レジスタを有する機能である。

【0003】

このレジスタ・リネーミング処理においては、例えばプロセッサに代表される構成により、物理レジスタとプログラム上で参照可能な論理レジスタが適切に割り当てられるようにレジスタ番号の比較を行なうことによって、レジスタ・バッファの識別を行ない、複雑なリネーミング処理を制御している。

【0004】

同時多重スレッド実行（以下、SMT—Simultaneous Multi-Threading—）のように、1つのプロセッサ内で複数のスレッドを同時に実行する構成の場合には、前述のレジスタ・リネーミング機能は同時実行数と同数の構成を有することになる（非特許文献2参照）。したがって、論理レジスタ数が増加することは、マイクロプロセッサ全体の構成の複雑さを増加させることになる。ここで、スレッドとは、論理演算を実行するための論理的な経路または筋道のことをいう。

【0005】

一方、ソフトウェア技術の向上により、レジスタ割り当ての柔軟性を高めることにより、プログラムの最適化によって性能向上を図ることが可能になってきている。例えば、マルチタスク処理を実現させるための1つのやり方としてのマルチスレッド化の場合を考える。処理の粒度を細かくしてスレッド化を行なった場合には、その処理内で必要とされるレジスタは少ない。しかし、あまり細かい粒度でのスレッド化は、スレッド間の処理の切り替え等のオーバーヘッド（overhead—限界を超える状態—）が現れて、スレッド化による効果を得ることができない場合がある。

【0006】

また、処理の粒度は簡単に細かく分割できるものではなく、処理の内容によっては処理の最適化の都合上、ある程度粒度が粗くなる場合もある。この場合、粒度が粗くなることにより、1つのスレッドで行なう処理が複雑になり、割り当てられるレジスタ数も増加することになる。以上のように、レジスタ数の増加は、ソフトウェアによる処理性能を向上させる長所を有しているが、ハードウェアの構成上の複雑さを増加させるという短所もある。

【0007】

図8は、従来のスーパースカラ方式によるマイクロプロセッサの構成例を示している（非特許文献1参照）。メモリから命令（instruction）をフェッチ（取り込むこと）する命令フェッチ部10と、命令フェッチ部10によりフェッチされた命令コードをデコードする命令デコード部20と、命令デコード部20からのコードによりレジスタ番号を一時的に保持する集中ウインド30と、レジスタ40と、プロセッサ50と、ALU（Arithmetic Logic Unit—演算論理回路—）60と、ストアユニット70と、ロードユニット80と、データキャッシュメモリ90より構成されている。説明を簡単にするため、図8ではデータパスの接続関係のみ示した。

【0008】

集中ウインド30はプロセッサ50からの対象となる物理レジスタの演算終了情報に基づいて発行可能な命令を発行し、レジスタ40からALU60、ストア

ユニット70、ロードユニット80へリネーミングされた物理レジスタにデータを供給し、アウト・オブ・オーダー (out-of-order) で演算実行される。演算結果は、リオーダーバッファ50へ送られ、イン・オーダー (in-order) で論理レジスタ番号に対応した結果を得ている。

【0009】

図9は、スーパースカラ方式で重要なレジスタ・リネーミング機構の一例を示しており、図9を用いてさらに詳細な動作の説明を行なう。レジスタ・リネーミング機構は、リオーダーバッファ2040と、物理レジスタ2010と、物理レジスタフリーリスト2020と、レジスタ・エイリアス・テーブル (register alias table) 2030より構成される。物理レジスタフリーリスト2020とレジスタ・エイリアス・テーブル2030は、図8に示される命令デコード20の中に組み込まれるものとする。リオーダーバッファ2040は、図8のリオーダーバッファ50をさらに詳細に示したものであり、命令番号、完了ビット、論理レジスタ番号、旧物理レジスタ番号より構成される。

【0010】

レジスタ・リネーミング方式では、論理レジスタ数よりも多い数の物理レジスタが存在する。命令がフェッチされ、デコードを行なう段階で命令コードのディスティネーション (行き先) で指定される論理レジスタ番号にその命令の結果が実際に格納される物理レジスタ番号が割り当てられる。その割り当ては、レジスタ・エイリアス・テーブル2030で示される。物理レジスタ番号の割り当ては、物理レジスタフリーリスト2020のレジスタ番号リストから読み出される。また、リオーダーバッファ2040には、命令番号を割り付けながら命令フェッチ順に命令が格納されており、先行命令が全て終了してリオーダーバッファから退去しているときに限り、後続命令を終了することができる。

【0011】

リオーダーバッファ2040は、命令番号1、2、3が割り付けられた3つの命令が発行された後、未終了のままで保持しており、命令番号3の命令が分岐命令であるものとする。リオーダーバッファ2040の完了ビットは「1」のときに命令実行中 (ビジー) であり、「0」のときに命令実行完了を示している。完了ビ

ットが示しているように、命令番号2の命令のみ命令実行完了の状態で、命令4がフェッチされ発行される状況からレジスタ・リネーミング機構の動作の流れを説明する。

【0012】

命令4はディスティネーション（行き先）として論理レジスタ番号2を使うものとする。その時、信号線1001を介して物理レジスタフリーリスト2020から物理レジスタ番号3をもらい、レジスタ・エイリアス・テーブル2030の当該物理レジスタ番号0を3に書き換え、信号線1003を介して以前の物理レジスタ番号をリオーダバッファ2040の旧物理レジスタ番号欄に書き込む。次に、命令番号1の命令実行が完了すると対応する完了ビットは「0」となり、論理レジスタ番号R1は以前の命令に未終了命令のないイン・オーダ（in-order）状態となり、その値はレジスタ・エイリアス・テーブル2030で示される論理レジスタ番号R1に対応する物理レジスタ番号10に格納される。

【0013】

リオーダバッファ2040の命令番号1に対応する旧物理レジスタ番号20は、命令番号1以前の命令で論理レジスタR1に書き込みを行なった物理レジスタ番号を表している。物理レジスタ番号10がR1に対応することになり、この旧物理レジスタ番号20は、信号線1004を介して物理レジスタフリーリスト2020に登録される。物理レジスタフリーリスト2020は、FIFO（First In First Out—先入れ先出し—）構造となっており、レジスタ・エイリアス・テーブル2030で物理レジスタ番号を割り当てる処理では登録された順に行なわれることになる。命令1が終了すると、命令2は既に実行終了しているので、リオーダバッファ2040から退去すると共に、命令1と同様に信号線1004を介して旧物理レジスタ番号が物理レジスタフリーリスト2020に登録される。物理レジスタ番号10, 0には論理レジスタR1, R2のイン・オーダ状態での値が記憶されている。

【0014】

命令4が命令3より早く終了すると、論理レジスタR2に対する物理レジスタ番号3に結果が格納され、信号線1005が指し示すレジスタに格納される。命

令 3 が未終了であるので、信号線 1 0 0 5 で指し示されたレジスタの値は先見状態となる。

【 0 0 1 5 】

ここで、命令 3 の分岐命令の実行において分岐予測ミスとなった場合、命令 4 の結果はキャンセルされることになる。その場合、リオーダバッファ 2 0 4 0 の命令 4 の項目で示される論理レジスタ R 2 で対応するレジスタ・エイリアス・テーブル 2 0 3 0 の物理レジスタ番号を、リオーダバッファ 2 0 4 0 の命令 4 の旧物理レジスタ番号 0 に書き戻しを行ない、リオーダバッファ 2 0 4 0 の命令 4 の項目を削除する。

【 0 0 1 6 】

したがって、論理レジスタ R 2 は、このレジスタ R 2 を書き換えたひとつ前の命令である命令 2 の演算結果が格納されている物理レジスタ 0 に対応される。物理レジスタ 3 は、物理レジスタフリーリスト 2 0 2 0 に書き戻される。

【 0 0 1 7 】

書き込みは全てレジスタ・エイリアス・テーブル 2 0 3 0 を介して論理レジスタに対応する物理レジスタ番号により行なわれる。値の読み出しに関しては、読み出し対象のレジスタ番号がリオーダバッファ 2 0 4 0 に格納（エントリ）されているか確認し、エントリされていなければレジスタ・エイリアス・テーブル 2 0 3 0 を介して論理レジスタに対応する物理レジスタ番号により行なわれる。

【 0 0 1 8 】

読み出し対象のレジスタ番号がエントリされている場合、リオーダバッファ 2 0 4 0 の完了ビットフィールドにより依存関係のある命令の実行修了を確認し、レジスタ・エイリアス・テーブル 2 0 3 0 を介して論理レジスタに対応する物理レジスタ番号により値の読み出しが行われる。レジスタ・リネーミング機構はこのような複雑な構成で実現されており、レジスタ・エイリアス・テーブル 2 0 3 0 およびリオーダバッファ 2 0 4 0 は連想方式により論理レジスタ番号の示すデータをやり取りすることになる。

【 0 0 1 9 】

したがって、論理レジスタ番号が増えることは、レジスタ・エイリアス・テ

ブル2030のサイズが大きくなり、その影響で物理レジスタ2010、物理レジスタフリーリスト2020のサイズを大きくする必要がある。また、リオーダバッファ2040の論理レジスタ番号のフィールドのサイズが増えることになる。その結果、レジスタ・リネーミング機構の構成がより複雑なものになりインプリメントを行なった場合、動作スピード上クリティカルパスを構成する要因となり動作目標であるスピードを実現する上でネックとなる可能性がある。

【0020】

図10は、2スレッド同時実行のマルチスレッド対応プロセッサの構成例を示している。物理レジスタ3400と物理レジスタ3700は説明の便宜上レジスタリードとレジスタライトの動作に分けて示してあるが、実体はひとつのレジスタであり、図10で示した物理レジスタ2010に対応するものとする。相互に依存関係のないプログラムAと、プログラムBを同時に実行可能とするため、命令フェッチユニット3100では2つのプログラムカウンタ3100と3120と命令キュー（待ち行列）3130で構成されており、命令フェッチは1サイクル毎にプログラムカウンタ3100と3120でのアクセス権を交代しながらフェッチを行なっている。

【0021】

一方のプログラムカウンタに対応する命令フェッチでキャッシュミス等によりフェッチできない場合、もう一方のプログラムカウンタにより連続して命令フェッチすることが可能となるように機能するものとする。命令キュー3130に格納された命令は、デコーダ3200内に設けられたレジスタ・エイリアス・テーブル3210、3220に各々対応する命令が振り分けられ、論理レジスタ番号に対応する物理レジスタ番号が割り当てられる。

【0022】

プログラムカウンタ3100にはレジスタ・エイリアス・テーブル3210、プログラムカウンタ3120にはレジスタ・エイリアス・テーブル3220が対応するものとする。デコーダ3200でデコードされた命令コードは、集中ウィンド3300に格納され物理レジスタ3400から演算に必要なデータを読み、実行ユニット3500に同時実行可能な限りの命令が投入される。実行ユニット

3500の演算結果は、物理レジスタ3700にレジスタ・エイリアス・テーブル3210、3220で示される論理レジスタ番号に対応する番号の物理レジスタに書き込まれ、リオーダバッファ3800で命令フェッチ順にその演算結果が確定される。

【0023】

リオーダバッファ3800には、2つのリオーダバッファ3810と3820で構成され、レジスタ・エイリアス・テーブル3210にリオーダバッファ3810が、レジスタ・エイリアス・テーブル3220にリオーダバッファ3820が対応し、各々独立に動作するものである。つまり、2スレッド同時実行のマルチスレッド対応プロセッサでは図9で説明したレジスタ・リネーミング機構が独立した状態で2つ存在することになり、nスレッド同時実行のマルチスレッド対応プロセッサとした場合、n個のプログラムカウンタとn組のレジスタ・リネーミング機構が存在することになる。したがって、図9で説明した論理レジスタ数増加に伴うレジスタ・リネーミング機構の構成上のネックは、マルチスレッド対応プロセッサにおいても同様に影響することになる。

【0024】

【特許文献1】

特開2002-163121号公報

【特許文献2】

特表2002-521762号公報

【特許文献3】

米国特許第6,247,106号明細書

【特許文献4】

特開平5-241830号公報

【非特許文献1】

ケネス・イーガ (Kenneth C. Yeager) 著「MIPS・R10000 スーパースカラマイクロプロセッサ (The MIPS 10000 Superscalar Microprocessor)」IEEEマイクロ、1996年4月、p. 28-33

【非特許文献2】

キース・ディフェンドルフ (Keith Diefendorff) 著「コンパックの
選択—アルファ用 SMT (Compaq Choose SMT for Alpha)」マイクロプロセッ
サ報告 (Cahners Microprocessor report)、1999年12月6日、Vol.
13、No. 16、p. 5-7

【0025】

【発明が解決しようとする課題】

キャッシングレジスタを設け、キャッシングレジスタとレジスタ本体との整合性を、後続命令のレジスタとキャッシング情報に従ってレジスタデータをロードまたはストアする命令を挿入して実行することにより、プログラム上参照可能な論理レジスタ数を増加させたとしても、アウト・オブ・オーダ (out-of-order) 実行等のような複雑な制御について従来と同様の構成により実現することを可能にする。

【0026】

【課題を解決するための手段】

上記目的を達成するため、本発明の基本構成としての第1構成に係るレジスタ・リネーミング機能を有するプロセッサは、命令を取り込む命令フェッチ部と、前記命令フェッチ部により取り込まれた前記命令から命令コードを復号するデコード部と、前記デコード部により復号された前記命令コードにより示されるレジスタ番号に対応するデータを保持するレジスタ部とを備え、デコードされた命令コードに含まれる論理レジスタ番号とその番号の物理レジスタ番号との対応をレジスタエイリアステーブルに保持すると共に前記物理レジスタ番号の割り当て可能な番号を物理レジスタフリーリストに格納して命令番号、処理状態、アーキテクチャレジスタ番号、古い物理レジスタ番号の対応関係を格納するリオーダバッファと前記物理レジスタフリーリストとを用いてレジスタエイリアステーブルの内容を順次書き換えるレジスタ・リネーミング機能を有するプロセッサにおいて、前記命令コードにより示されるレジスタ番号に対応するデータを保持するレジスタ本体と、前記レジスタ本体に保持された内容の一部をキャッシングするキャッシングレジスタと、前記命令フェッチ部からの命令により前記キャッシングレジスタにより保持されている論理レジスタ番号やキャッシングレジスタ番号を含

む内部命令の状態に関する情報を保持する内部命令情報保持部と、前記命令フェッチ部からの命令をプリデコードした命令コードと前記内部命令情報保持部により保持された前記内部命令の状態に関する情報とを比較して内部命令を挿入するか否かを判定する命令挿入判定部と、前記命令挿入判定部により内部転送命令を挿入するべきものと判定されたときに前記キャッシングレジスタと前記レジスタ本体との間で内部データの転送を行なうレジスタ転送命令を発行するレジスタ転送命令発行部と、を備えることを特徴とする。

【 0 0 2 7 】

本発明の第 2 構成に係るレジスタ・リネーミング機能を有するプロセッサは、上記第 1 構成のプロセッサにおいて、前記レジスタ本体はプログラム上で参照可能な論理レジスタより構成され、前記キャッシングレジスタは前記論理レジスタの一部を保持する内部レジスタより構成され、前記レジスタ転送命令発行部は前記論理レジスタと前記内部レジスタとにより使用される内部レジスタ番号に変換する変換部と前記論理レジスタと前記内部レジスタとの間でデータ転送を行なうプロセッサ内部命令コードと同じ形式のコードを生成するコード生成部とより構成されることを特徴とする。

【 0 0 2 8 】

本発明の第 3 構成に係るレジスタ・リネーミング機能を有するプロセッサは、上記第 1 構成に係るプロセッサにおいて、前記命令フェッチ部からの命令を命令コードにプリデコードするプリデコード部をさらに備えることを特徴とする。

【 0 0 2 9 】

本発明の第 4 構成に係るレジスタ・リネーミング機能を有するプロセッサは、第 3 構成に係るプロセッサにおいて、前記内部命令情報保持部、前記プリデコード部、前記命令挿入判定部、前記レジスタ転送命令発行部によりロードレジスタ命令とストアレジスタ命令を挿入して命令を発行するレジスタ命令挿入ユニットが構成されていることを特徴とする。

【 0 0 3 0 】

本発明の第 5 構成に係るレジスタ・リネーミング機能を有するプロセッサは、第 4 構成に係るプロセッサにおいて、前記レジスタ命令挿入ユニットは、前記内

部命令情報を保持する内部命令保持部と、前記命令フェッチ部より供給される命令コードからレジスタ番号情報を取り出すプリデコード部と、このプリデコード部からの前記レジスタ番号情報と前記内部命令保持部が保持する内部命令情報のTAG領域に格納される論理レジスタ番号の比較を行なって挿入命令のレジスタ番号を生成する前記命令挿入判定部としての挿入命令レジスタ番号生成部と、この挿入命令レジスタ番号生成部により生成された前記挿入命令のレジスタ番号に基づいてロード／ストアレジスタ命令を発行する前記レジスタ転送命令発行部としてのロード／ストアレジスタ命令発行部と、を備えることを特徴とする。

【0031】

本発明の第6構成に係るレジスタ・リネーミング機能を有するプロセッサは、第5構成に係るプロセッサにおいて、前記命令フェッチ部と前記命令デコード部との間に、前記ロード／ストアレジスタ命令発行部により発行された命令に基づいて、前記命令フェッチ部より供給される命令の前に前記ロード／ストアレジスタ命令発行部からの命令を付加して前記命令デコード部に供給する命令挿入制御部をさらに備えることを特徴とする。

【0032】

【発明の実施の形態】

以下、添付図面を参照しながら本発明に係るレジスタ・リネーミング機能を有するプロセッサの実施形態について詳細に説明する。まず、図1に従って本発明の基本的な原理を含む第1実施形態に係るプロセッサについて説明する。

【0033】

図1に示すように、第1実施形態に係るプロセッサは、命令を取り込む命令フェッチ部1と、命令フェッチ部1により取り込まれた前記命令から命令コードを復号する命令デコード部2と、命令デコード部2により復号された命令コードを保持するレジスタ部3とを備えている。

【0034】

さらに、第1実施形態に係るプロセッサは、デコードされた命令コードに含まれる物理レジスタ番号とその番号の物理レジスタ内の命令との対応をレジスタエイリアステーブルに保持すると共に前記物理レジスタ番号の割り当てを物理レジ

スタフリーリストに格納して命令番号、処理状態、アーキテクチャレジスタ番号、古い物理レジスタ番号の対応関係を格納するリオーダバッファ2040と物理レジスタフリーリストとを用いてレジスタエイリアステーブルの内容を順次書き換えるレジスタ・リネーミング処理部2030を有している。リオーダバッファ2040を用いるレジスタ・リネーミング処理部2030の機能については図9を用いて説明した従来のものと略々同じであるので、重複説明を省略すると共に第2実施形態の説明においてさらに詳細に説明する。

【0035】

このような基本構成を有するプロセッサにおいて、前記命令コードを保持するレジスタ本体3と、レジスタ本体3に保持された内容の一部をキャッシングするキャッシングレジスタ4と、レジスタ命令を挿入するためのレジスタ命令挿入ユニット5とを備えている。

【0036】

レジスタ命令挿入ユニット5は、命令フェッチ部1からの命令によりキャッシングレジスタ4により保持されている論理レジスタ番号やキャッシングレジスタ番号を含む内部命令の状態に関する情報を保持する内部命令情報保持部6と、前記命令フェッチ部からの命令をプリデコードするプリデコード部7と、プリデコード部7によりプリデコードされた命令コードと内部命令情報保持部6により保持された内部命令の状態に関する情報とを比較して内部命令を挿入するか否かを判定する命令挿入判定部8と、命令挿入判定部8により内部転送命令を挿入すべきものと判定されたときに前記キャッシングレジスタ4と前記レジスタ本体3との間で内部データの転送を行なうレジスタ転送命令を発行するレジスタ転送命令発行部9と、を備える。

【0037】

以上のような第1実施形態に係るプロセッサによれば、キャッシングレジスタ4を設け、このキャッシングレジスタ4とレジスタ本体3との整合性を、後続命令のレジスタとキャッシング情報とに従ってレジスタデータをレジスタ命令挿入ユニット5によってロードあるいはストアする命令を挿入して実行することにより、プログラム上参照可能な論理レジスタ数を増加させても、アウト・オブ・オ

ーダ (out-of-order) 実行等のような複雑な制御も従来通りの構成で可能にすることができる。

【 0 0 3 8 】

次に、本発明に係るレジスタ・リネーミング機能を有するプロセッサの第2実施形態について図2ないし図6を参照しながら説明する。この第2実施形態に係るプロセッサは、レジスタ本体3とキャッシングレジスタ4との動作がより詳細に説明される。

【 0 0 3 9 】

ソフトウェア技術の向上により、プログラムをマルチストリーム化し独立に実行できたとしても、プログラムの最小単位では、直前命令の処理結果に依存して命令実行されることはプログラムの特性上から避けられない。このことは、使用される論理レジスタに局所性が生じることを意味する。

【 0 0 4 0 】

使用される論理レジスタに局所性があることから、全体の論理レジスタの一部をキャッシングし、キャッシングされたレジスタのみを対象にレジスタ・リネーミング等の制御を行なえば論理レジスタ数増加に伴うハードウェア構成上の複雑度は軽減される。しかし、通常のデータキャッシュのような構成にした場合、キャッシュミスした場合にはリフィルする必要がある、リフィル処理によってパイプラインが長い期間ストールし性能低下を招くことになる。

【 0 0 4 1 】

図2には、上記の問題も含めて回避した、レジスタファイルのキャッシング機構を有するプロセッサ構成が示されている。図2において、図8と同一符号を付した構成要素は、同一または相当する機能を有するものとして、重複説明は省略する。本発明の第2実施形態に係るプロセッサにおいては、図8に示されたレジスタ40をRAM400として設置してレジスタの本体とし、集中ウインド30の下にキャッシングレジスタ300を配置する。キャッシングレジスタ300はロード/ストアレジスタ命令挿入ユニット100から供給される命令によりRAM400とのデータ交換を行なう。動作としてはデータキャッシュメモリ90に対するロード/ストア命令と同様のデータ転送動作を行なう。

【0042】

ロード／ストアレジスタ命令挿入ユニット100は、TAG110とロード／ストアレジスタ命令発行装置120より構成される。TAG110は、命令フェッチ10の下に配置され、命令フェッチ10から供給される命令コード中のソースレジスタナンバーRs、Rt、ディスティネーションレジスタナンバーRdによりロード／ストアレジスタ命令発行装置120に対し、命令を挿入の制御を行なう。

【0043】

TAG110には、命令フェッチ10からの命令コード情報により、キャッシングレジスタ300にレジスタキャッシングされている論理レジスタ番号、キャッシングされているキャッシングレジスタ番号、キャッシングデータがRAM400との間で好ましくない（dirty—ダーティー—）状態であるかどうかの情報を持っている。図3にはTAG110の構成を、図4にはTAG110の制御フローを示す。図2において、タグ（Tag.）領域には図3のフローに従って論理レジスタ番号がセットされる。キャッシングレジスタ番号領域（図中ではC_Reg. No. とする）には、TAGにセットされた論理レジスタ番号のデータがセットされるキャッシングレジスタ300のレジスタ番号がセットされる。Dは、キャッシングレジスタ300のデータがダーティーであるかどうかのフラグであり、D=1の場合はダーティーとする。TAG110の初期状態は、キャッシングレジスタ番号と同じ論理レジスタ番号が割り当てられ、全てD=0という状態であるとする。

【0044】

図4の制御フローに従って動作を説明する。始めに、ソースレジスタ番号とTAGを比較し、一致しなければ、TAG110中にソースレジスタ番号をTAGに割り当て、対応するキャッシングレジスタ番号に対するロードレジスタ命令挿入をロード／ストアレジスタ命令挿入ユニット120に対して要求する。また、対応するDビット（bit）を0にクリアする。ソースレジスタ番号とTAGを比較し一致した場合には、Dビットをチェックし、D=1の場合、対応するC_Reg番号のデータがダーティーな状態であることを示しているので、ストアレジ

スタ命令挿入をロード／ストアレジスタ命令挿入ユニット200に対して要求する。図4のフローチャートではソースレジスタRs, Rtの順で前記チェックを行なっているが、同時に実行しても構わない。

【0045】

ソースレジスタに対する処理後、ディスティネーションレジスタ番号とTAGを比較し、一致しなければ、TAG400中にディスティネーション（行き先）レジスタ番号をTAGに割り当て、D bitを0にクリアし、ストアレジスタ命令挿入をロード／ストアレジスタ命令挿入ユニット200に対して要求する。ディスティネーションレジスタ番号とTAGが一致した場合は、一致したキャッシングレジスタ（C_Reg.）番号のデータを書き換えることになるのでD=1をセットしてダーティーな状態としてセットし終了する。

【0046】

TAGの置き換えは、一般的に知られているLRU(least-recently used)等の置換アルゴリズムを使って最も使用頻度の少ないキャッシングレジスタ番号のデータを論理レジスタ番号が示すRAM400の番地に戻し、新規にエントリされる論理レジスタ番号のそのキャッシングレジスタ番号を割り当てて置き換えを行なう。

【0047】

以上のような処理の流れにより生成された命令挿入要求によって、ロード／ストアレジスタ命令挿入ユニット100は、対象となる命令の前に要求のあった挿入命令をストアレジスタ命令、ロードレジスタ命令の順で挿入して、集中ウインド30に供給している。

【0048】

これ以降の動作は従来のプロセッサと同様の動作であり、キャッシングレジスタ300とレジスタ本体であるRAM400との整合性はレジスタロード命令／レジスタストア命令挿入により整合性を取っている。したがって、長い期間パイプラインストールすることがなく、レジスタネーミング処理や、アウト・オブ・オーダー（out-of-order）実行といった制御を従来のままの構成で、より多くの論理レジスタ数を扱うことが可能となる。図5は、従来例の図9で示したレジスタ

・リネーミング機構の構成において、本第2実施形態であるキャッシングレジスタを設けたレジスタ・リネーミング機構の構成例を示している。

【0049】

図5において、図9と同一符号の構成要素は同一または相当する機能を有するものである。レジスタ・エイリアス・テーブル2030は、図9の例では論理レジスタ番号と物理レジスタ番号との対応を取るテーブルであるが、図5におけるレジスタ・エイリアス・テーブル2030では、キャッシングレジスタ番号と物理レジスタ番号との対応を取るテーブルとなっている。ただし、キャッシングレジスタ番号と論理レジスタ番号が差し変わっただけで、レジスタ・エイリアス・テーブル2030で扱うキャッシングレジスタ番号と従来例で扱う論理レジスタ番号の数は全く同一の番号である。

【0050】

したがって、従来のレジスタ・エイリアス・テーブル2030と図5で示すレジスタ・エイリアス・テーブル2030は同一のものである。同様に、図5におけるプロセッサ2040および物理レジスタフリーリスト2020も扱う番号が論理レジスタ番号からキャッシングレジスタ番号に差し替えただけで全く同一の構成となっている。

【0051】

図5において、レジスタ・エイリアス・テーブル2030、プロセッサ2040、物理レジスタフリーリスト2020、物理レジスタ2010で、図9で説明したレジスタ・リネーミング機構と同じもので構成されている。これらの動作については、従来例の図9の説明される動作と全く同じ動作となるので説明を省略する。本第2実施形態では、従来のレジスタ・リネーミング機構の構成加えて、論理レジスタ4000、TAG4100、LRUカウンタ4200により構成される。

【0052】

TAG4100は、図3に示したTAG110に対応するもので、論理レジスタ番号を示すTAGフィールド4110と、ダーティービット4120とキャッシングレジスタ番号、フィールド4130に加えてキャッシングレジスタエント

リビット4 1 4 0で構成されている。

【0 0 5 3】

キャッシングレジスタエントリビット4 1 4 0は現在キャッシングレジスタにエントリされている論理レジスタ番号を示すビットで、TAGフィールド4 1 1 0の論理レジスタ番号に対応するキャッシングレジスタエントリビット4 1 4 0が「1」の場合、キャッシングレジスタにエントリされていることとする。

【0 0 5 4】

図3で説明したロードレジスタ命令が発行された場合、ロードレジスタ命令で指定される論理レジスタ番号に一致するキャッシングレジスタエントリビット4 1 4 0が「1」にセットされる。このとき、ロードレジスタ命令に割り当てられるキャッシングレジスタ番号は最もアクセス頻度の少ないキャッシングレジスタ番号が割り当てられる。このとき、キャッシングレジスタエントリビット4 1 4 0が1でかつキャッシングレジスタ番号、フィールド4 1 3 0のキャッシングレジスタ番号にマッチした論理レジスタ番号、つまりキャッシングレジスタにエントリされており、置き換え対象となる論理レジスタ番号のダーティービット4 1 2 0が「1」でダーティーな場合、対象となる割り当てを行なうキャッシングレジスタ番号、に対するストアレジスタ命令が作成され、対応するダーティービット4 1 2 0が0にクリアされる。

【0 0 5 5】

ストアレジスタ命令、ロードレジスタ命令の順で発行され、ストアレジスタ命令は、通常のフェッチされるレジスタを参照するのみの命令と同様に、レジスタ・エイリアス・テーブル2 0 3 0のキャッシングレジスタ番号、に対応する物理レジスタ番号を参照し、置き換え対象の論理レジスタ番号へストアするという命令として作成され、プロセッサ2 0 4 0に該当するキャッシングレジスタ番号がエントリされていないか確認する手順を踏んで実行される。その命令が実行されると、信号線4 3 1 0を經由して物理レジスタ2 0 1 0から論理レジスタ4 0 0 0の指定先へとデータがストアされることになる。

【0 0 5 6】

後続のロードレジスタ命令も通常のフェッチされるレジスタ書き換えを行なう

命令と同様に、レジスタ・エイリアス・テーブル 2 0 3 0 のキャッシングレジスタ番号に対応するテーブルに物理レジスタフリーリスト 2 0 2 0 より割り当てを行なう物理レジスタ番号を割り当てて、プロセッサ 2 0 4 0 にエントリされて実行されると、信号線 4 3 0 0 を経由して、論理レジスタ 4 0 0 0 から物理レジスタ 2 0 1 0 へレジスタ・エイリアス・テーブル 2 0 3 0 で示される物理レジスタ番号へデータがロードされる。

【 0 0 5 7 】

以上のような動作によって、キャッシングレジスタの入れ替えが行なわれる。図 4 に示されたフローチャートにより説明したように、TAG 4 1 0 0 でソースレジスタナンバー R_s , R_t に該当する論理レジスタがキャッシングレジスタにエントリされていれば、ストアレジスタ命令、ロードレジスタ命令が挿入されることなく、通常のフェッチされた命令のみが従来例と同様の動作で実行され続ける。

【 0 0 5 8 】

キャッシングレジスタの入れ替え動作時の TAG 4 1 0 0 で行なわれるキャッシングレジスタ番号の割り当ては、前述したように本実施例では最もアクセス頻度の少ないキャッシングレジスタ番号を割り当てることとする。その実現は、例えばカウンタを用いた LRU アルゴリズムにより簡単に実現できる。LRU カウンタ 4 2 0 0 は、TAG 4 1 0 0 で通常のフェッチされた命令のソースレジスタナンバー R_s , R_t のキャッシングレジスタへのエントリチェック時、ヒットしたキャッシングレジスタ番号、に対応するカウンタを 0 にクリアし最近アクセスされたことを記憶する。

【 0 0 5 9 】

また、ヒットしたカウンタのクリア前カウントより小さいカウンタは、1 つインクリメントする。それ以外のカウンタはそのままとすることにより、各キャッシングレジスタ番号、にエントリされている論理レジスタ番号のアクセス頻度が相対的に示されることになり、最も値の大きいものが最もアクセス頻度の少ないキャッシングレジスタ番号となる。

【 0 0 6 0 】

仮に、キャッシングレジスタ数の半分のレジスタ数のみを使用するプログラムが実行され、キャッシングレジスタがヒットし続けると、残り半分のカウンタはオーバーフローすることになるが、その場合はキャッシングレジスタ番号の小さい順に割り当てるなどのプライオリティ（優先度）により割り当てを行えば良い。また、オーバーフロー以外でもカウンタの値が最大なものが複数存在した場合でも同様方法で対応可能である。

【 0 0 6 1 】

図 6 には、図 1 0 で説明した、2 スレッド同時実行のマルチスレッド対応プロセッサの構成例に本実施例を組み込んだ場合の構成例を示す。従来例と同様に物理レジスタ 3 4 0 0 と物理レジスタ 3 7 0 0 は説明の便宜上レジスタリードとレジスタライトの動作で分けて示してあるが、実体はひとつのレジスタであり、図 5 で示した物理レジスタ 2 0 1 0 に対応するものであるとする。

【 0 0 6 2 】

デコーダ 3 2 0 0 中には、2 つのロード／ストアレジスタ命令挿入ユニット 5 2 1 0 と 5 2 2 0 と、実行ユニット 3 5 0 0 中に論理レジスタ 5 3 0 0 を加えている。ロード／ストアレジスタ命令挿入ユニット 5 2 1 0 と 5 2 2 0 は各々レジスタ・エイリアス・テーブル 3 2 1 0 とレジスタ・エイリアス・テーブル 3 2 2 0 に対応し、その動作の関係は図 5 で説明した動作と全く同様である。また、ロード／ストアレジスタ命令挿入ユニット 5 2 1 0 と 5 2 2 0 がキャッシングの対象とする論理レジスタは、各々論理レジスタ 5 3 0 0 に実装される全レジスタ番号を対象とし特に制限されるものではない。

【 0 0 6 3 】

ストアレジスタ命令がロード／ストアレジスタ命令挿入ユニット 5 2 1 0 または 5 2 2 0 から発行された場合、物理レジスタ 3 4 0 0 から論理レジスタ 5 3 0 0 ヘデータ転送される。ロードレジスタ命令がロード／ストアレジスタ命令挿入ユニット 5 2 1 0 または 5 2 2 0 から発行された場合には、論理レジスタ 5 3 0 0 から物理レジスタ 3 7 0 0 ヘデータが転送される。

【 0 0 6 4 】

従来例で示したようにプログラム A、プログラム B 相互に依存関係のない処理

である。仮に、プログラムA、プログラムBに共通の論理レジスタ番号が割り当てられたとすると、本実施例においてはストアレジスタ命令またはロードレジスタ命令のデータ転送処理で同一論理レジスタ番号に対するアクセスが発生してしまう。この問題は、本実施形態により増加した論理レジスタによりソフトウェアにより各スレッドプログラムへのレジスタ割り当ての最適化と各スレッドの実行スケジューリングによるソフト上の最適化により回避されるものとする。

【 0 0 6 5 】

この第2実施形態によるプロセッサによれば、論理レジスタの増加がソフトウェアによる最適化を容易にすることになる。また、第2実施形態では、インプリメントを行なった場合に問題となる、動作スピード上クリティカルとなる要因を分散させることになる。

【 0 0 6 6 】

実行ユニット3500中にある演算ユニット等が演算処理のために必要なデータをリードし演算処理後に結果をライトするといった処理に比べ、単にデータ転送するのみのストアレジスタ命令またはロードレジスタ命令の処理はシンプルであり動作スピード上問題とはならない。また、従来例において論理レジスタ数が仮に4倍になった場合には、各レジスタ・エイリアス・テーブル、物理レジスタのサイズがリニアに（正比例して）増加すると共に、プロセッサの論理レジスタ番号フィールドの増加により、プロセッサ3800とデコーダ3200を跨ぐ必要な信号線が飛躍的に増加することになる。

【 0 0 6 7 】

この第2実施形態では、各レジスタ・エイリアス・テーブル、物理レジスタ、プロセッサ構成はそのまま、論理レジスタを実行ユニットに加える増加と、デコーダ3200内でのロード／ストアレジスタ命令挿入ユニットはレジスタ・エイリアス・テーブルとの関係の内部信号に限定されるため、従来の動作スピード上クリティカルとなる問題を解決すると共に、論理レジスタ数を容易に増加することが可能となる。

【 0 0 6 8 】

次に、図7を用いて本発明の第3実施形態に係るレジスタ・リネーミング機能

を有するプロセッサについて説明する。第3実施形態に係るプロセッサは、図4のフローチャートに示した第2実施形態のプロセッサの制御フローを実現するロード／ストアレジスタ命令挿入ユニット100の構成の一例を示している。

【0069】

図7において、TAG110と、命令フェッチ10より供給される命令コードからレジスタナンバー情報を取り出すプリデコード装置130と、このプリデコード装置130からのレジスタナンバー情報と、TAG110のタグ(Tag.)領域に格納される論理レジスタ番号の比較を行い、挿入命令のレジスタナンバーを生成する挿入命令レジスタ番号生成装置140と、ロード／ストアレジスタ命令発行装置120と、命令挿入制御装置150とより構成される。

【0070】

プリデコード装置130は、命令フェッチ10より供給される命令よりレジスタナンバー情報を取り出し、挿入命令レジスタ番号生成装置140へ供給する。挿入命令レジスタ番号生成装置140は、供給されたレジスタナンバー情報によりTAG110のタグ(Tag.)領域に格納される論理レジスタ番号の比較を行う。レジスタ番号情報の比較は、ディスティネーションレジスタナンバーRd、ソースレジスタナンバーRs、Rt毎にそれぞれTAG110のタグ(Tag.)領域の全ての論理レジスタ番号と連想方式により比較を行なう。

【0071】

ソースレジスタナンバーRs、Rtの比較を行ない、一致した場合、対応するTAG110のDビット(bit)を参照し、D=1でダーティーな場合には、信号線111、112よりストアレジスタ命令のレジスタ番号情報がロード／ストアレジスタ命令発行装置120に供給される。また、ストアレジスタ命令のレジスタ番号情報を供給と同じタイミングで、信号線145、146を介して、TAG110のDビットをクリアする。D=0の場合には何も行なわない。

【0072】

ソースレジスタナンバーRs、Rtの比較が不一致の場合には信号線141、142をよりロードレジスタ命令のレジスタ番号情報がロード／ストアレジスタ命令発行装置120に供給され、信号線113よりロードすべきキャッシングレ

ジスタ番号。がロード／ストアレジスタ命令発行装置120に供給される。また、ロードレジスタ命令のレジスタ番号情報の供給と同じタイミングで、信号線145, 146を介しTAG110のDビットをクリアする。

【0073】

ディスティネーションレジスタナンバーRdの比較では、一致した場合、信号線147を介して一致したタグ(Tag.)領域の対応するDビットを1にセットし、ダーティーな状態になることをセットする。ディスティネーションレジスタナンバーRdの比較が不一致の場合には、LRU (least-recently used) 等の置換アルゴリズムでTAG110の追い出すエントリを決定し、決定されたキャッシングレジスタ番号。に対応するレジスタ番号情報が信号線111, 112を介してストアレジスタ命令のレジスタ番号情報としてロード／ストアレジスタ命令発行装置120に供給される。また、信号線145を介してディスティネーションレジスタナンバーRdが置換対象となったTAG110のタグ(Tag.)領域に供給される。

【0074】

ロード／ストアレジスタ命令発行装置120は、命令コードのオペコード部にロードレジスタ(Ld. Reg.) 命令、ストアレジスタ(St. Reg.) 命令を示すコードが埋め込まれたフェッチされる命令コードと同形式のコードがROMまたはPLA等の構成によりコードを有しており、挿入命令レジスタ番号生成装置140より供給されるレジスタ番号情報を組み込むことで、実際に発行する挿入命令コードを作成する。

【0075】

また、ロードレジスタ(Ld. Reg.) 命令、ストアレジスタ(St. Reg.) 命令の各々に対応するレジスタ番号情報が供給されることにより、命令発行の必要の有無を判断し、作成した命令コードを命令挿入制御装置150に命令発行する。ロードレジスタ(Ld. Reg.) 命令に対応するレジスタ情報が供給されなければ、ロードレジスタ(Ld. Reg.) 命令は発行されず、ストアレジスタ(St. Reg.) 命令に対応するレジスタ情報が供給されなければ、ストアレジスタ(St. Reg.) 命令は発行されない。

【 0 0 7 6 】

命令挿入制御装置 1 5 0 は、ロード／ストアレジスタ命令発行装置 1 2 0 より命令発行が行われると、命令フェッチ 1 0 より供給される命令の前にロード／ストアレジスタ命令発行装置 1 2 0 からの命令を付けて命令デコード 20 に供給する。以上の構成により、図 4 に示した制御フローにおける動作を実現することができる。

【 0 0 7 7 】

【発明の効果】

プログラム上で参照可能な論理レジスタ数を扱うプロセッサにおける制御の複雑度を抑えることができると共に、長い期間のパイプラインストールを発生することなく、多くの論理レジスタ数を扱えることができ、ソフトウェア最適化による性能向上が可能になる。

【図面の簡単な説明】

【図 1】

本発明の基本概念としての第 1 実施形態に係るマイクロプロセッサの構成を示すブロック図である。

【図 2】

本発明の第 2 実施形態によるレジスタファイルを有するマイクロプロセッサの構成を示す機能ブロック図である。

【図 3】

図 2 における T A G 1 1 0 の詳細な構成を示す説明図である。

【図 4】

同じく図 2 の T A G 1 1 0 での制御フローを示すフローチャートである。

【図 5】

本発明の第 2 実施形態に係る実施例のレジスタファイルを有するマイクロプロセッサにおけるレジスタ・リネーミング機構の構成を示す説明図である。

【図 6】

本発明の第 2 実施形態におけるマルチスレッド対応のマイクロプロセッサの構成例を示す機能ブロック図である。

【図 7】

図 4 の制御フローを実現する構成としての第 3 実施形態に係るマイクロプロセッサのロード／ストアレジスタ命令挿入ユニットを示すブロック図である。

【図 8】

従来のマイクロプロセッサの一例を示す機能ブロック図である。

【図 9】

従来のマイクロプロセッサにおけるレジスタ・リネーミング機構の構成を示す説明図である。

【図 1 0】

従来のマルチスレッド対応のマイクロプロセッサの構成例を示す機能ブロック図である。

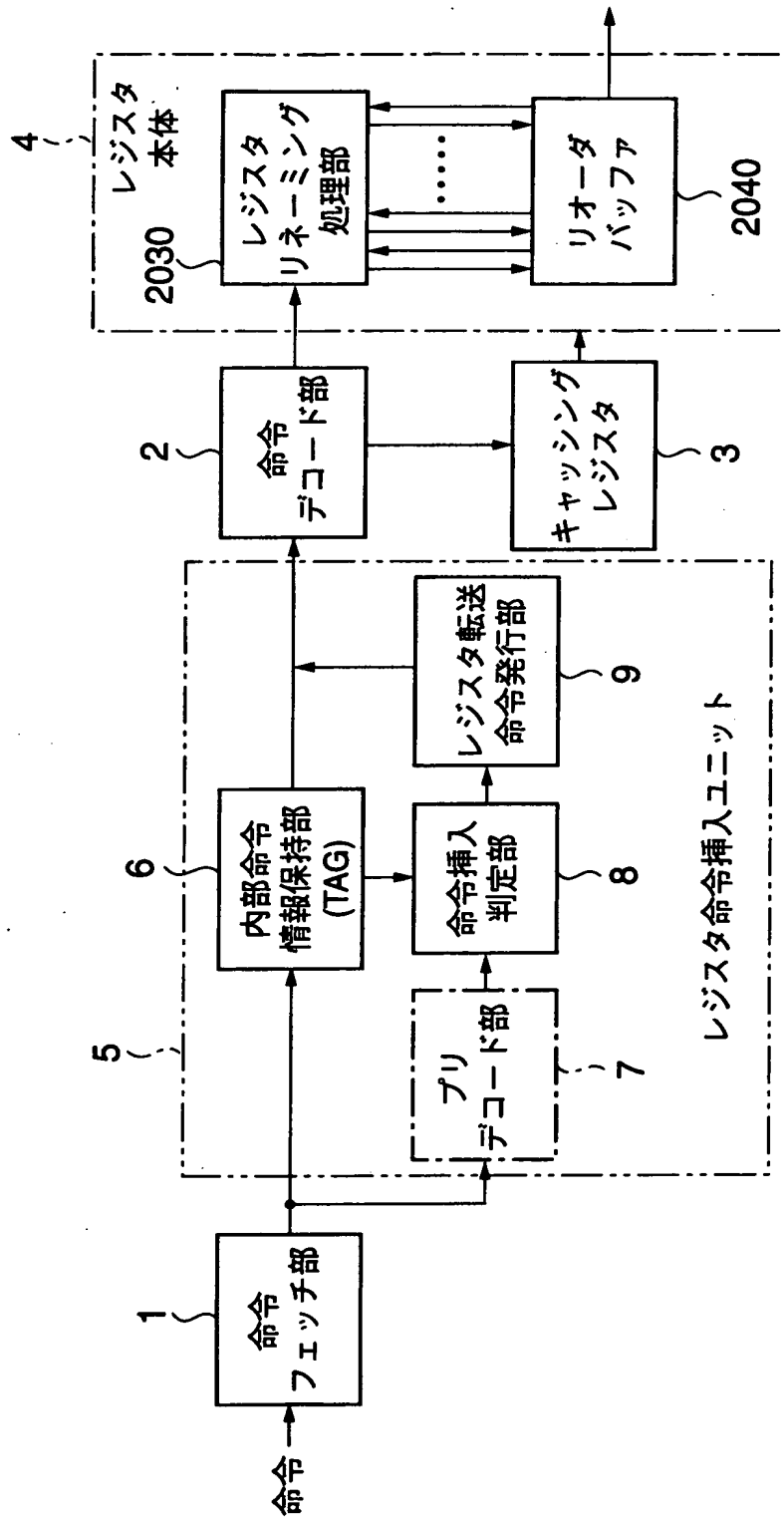
【符号の説明】

- 1 命令フェッチ部
- 2 命令デコード部
- 3 キャッシングレジスタ
- 4 レジスタ本体
- 5 レジスタ命令挿入ユニット
- 6 内部命令情報保持部 (TAG)
- 7 プリデコード部
- 8 命令挿入判定部
- 9 レジスタ転送命令発行部
- 10 命令フェッチ部
- 20 命令デコード部
- 30 集中ウインド
- 40 レジスタ本体
- 50 プロセッサ
- 60 ALU
- 70 ストアユニット
- 80 ロードユニット

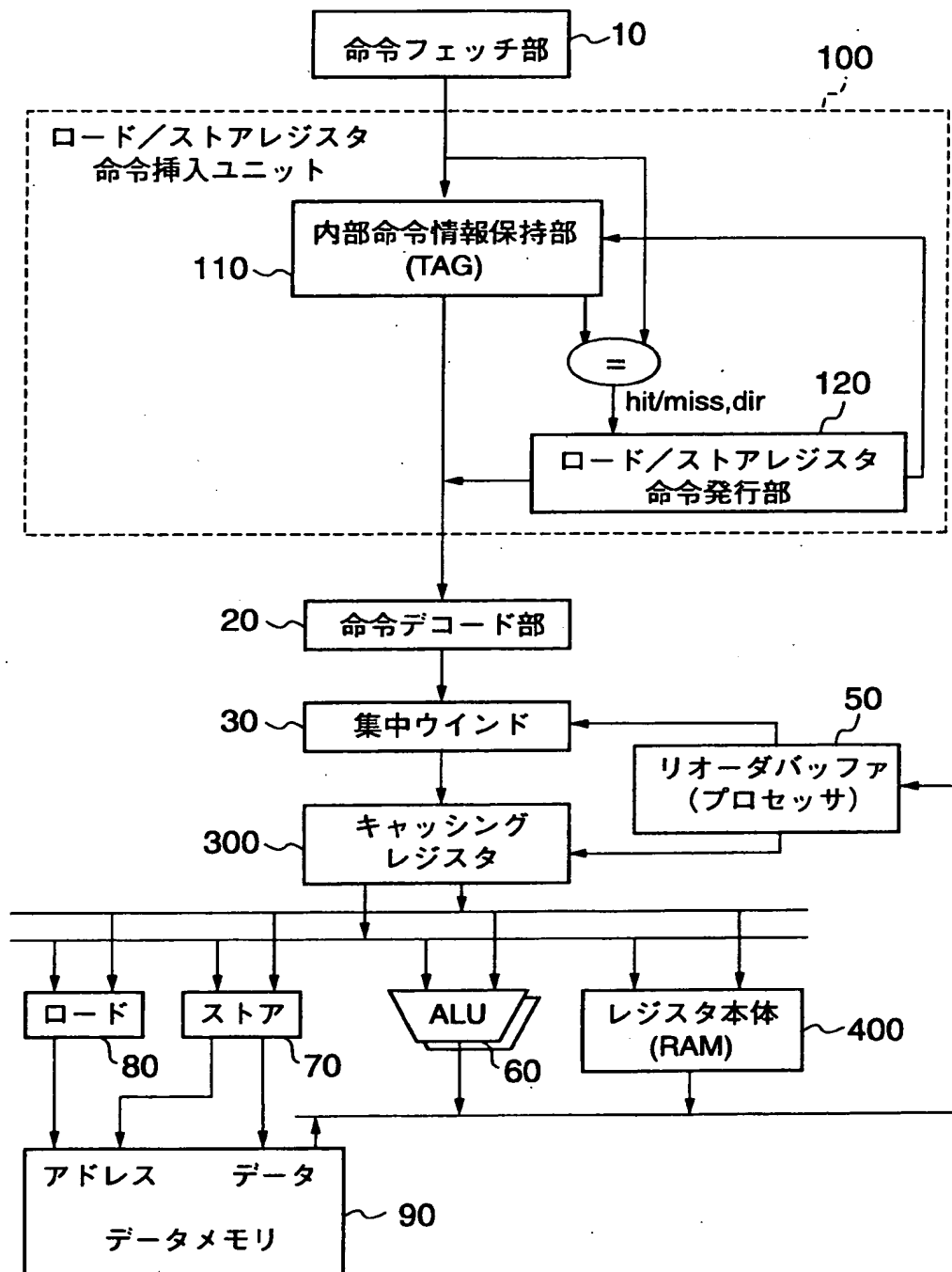
- 9 0 データキャッシュメモリ
- 1 0 0 (ロード／ストア) レジスタ命令挿入ユニット
- 1 1 0 内部命令情報保持部 (T A G)
- 1 2 0 レジスタ転送命令発光部 (ロード／ストアレジスタ命令発行部)
- 1 3 0 プリデコード部
- 1 4 0 命令挿入判定部 (挿入命令レジスタ番号生成部)
- 1 5 0 命令挿入制御部
- 3 0 0 キャッシングレジスタ
- 4 0 0 R A M (レジスタ本体)
- 2 0 3 0 レジスタ・リネーミング処理部
- 2 0 4 0 リオーダーバッファ

【書類名】 図面

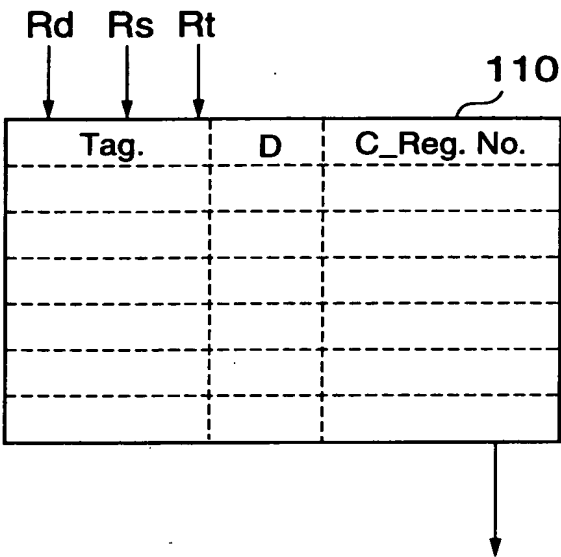
【図 1】



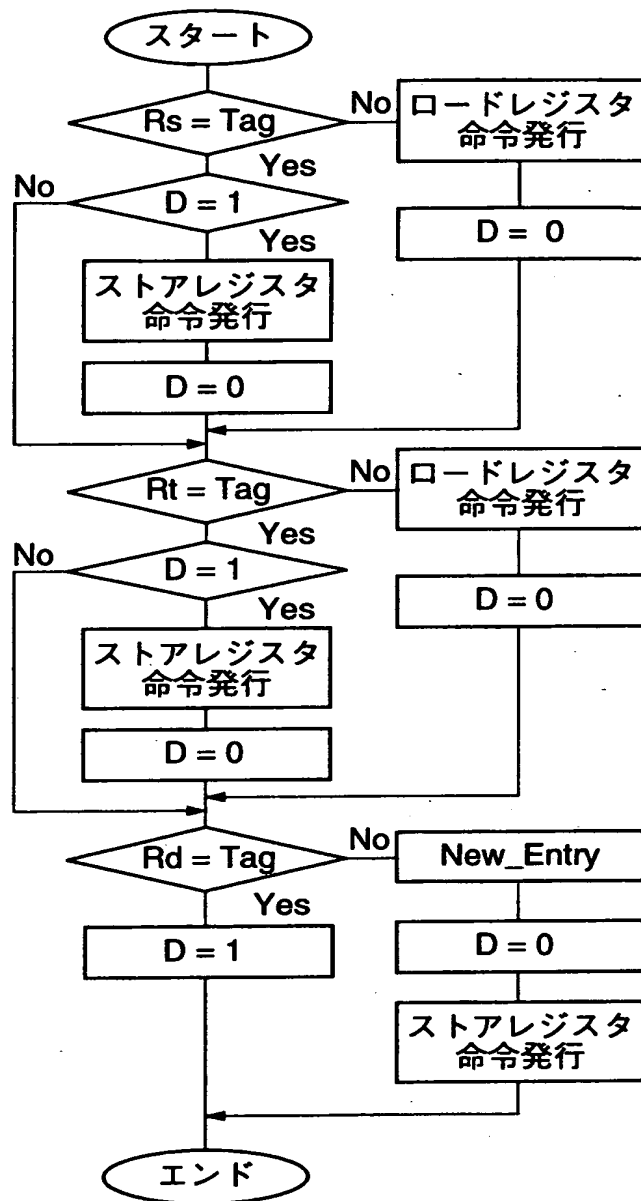
【図 2】



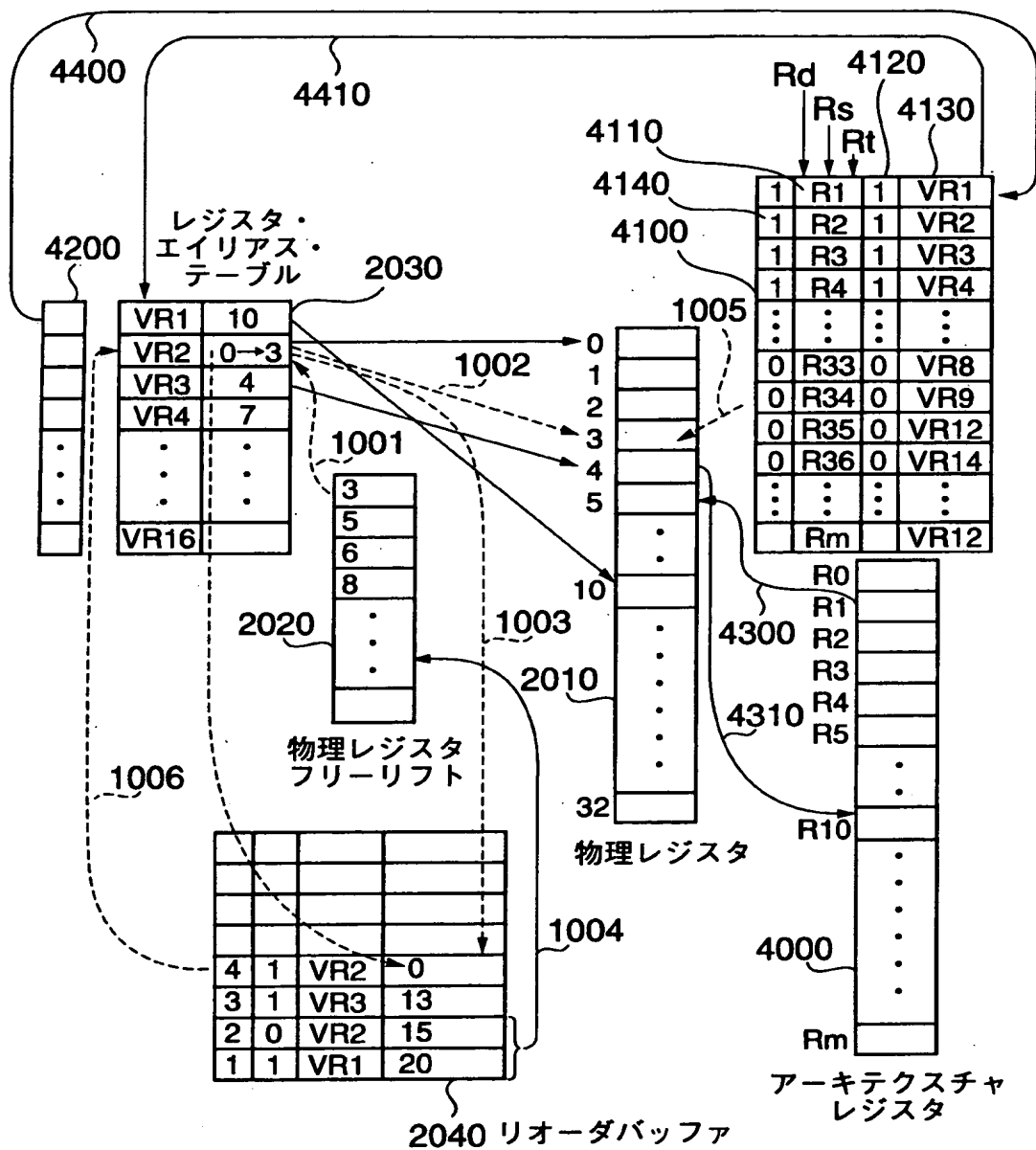
【図 3】



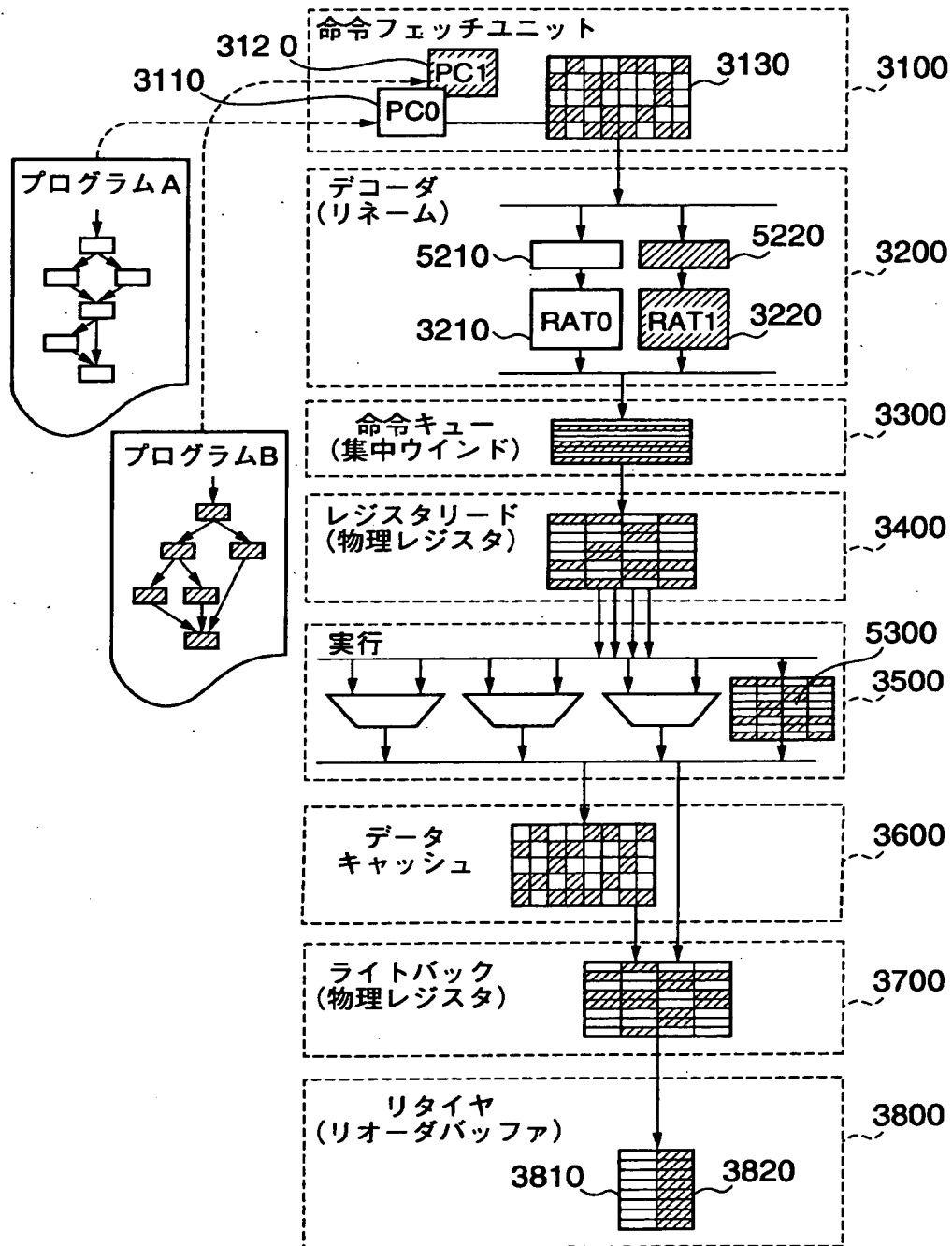
【図 4】



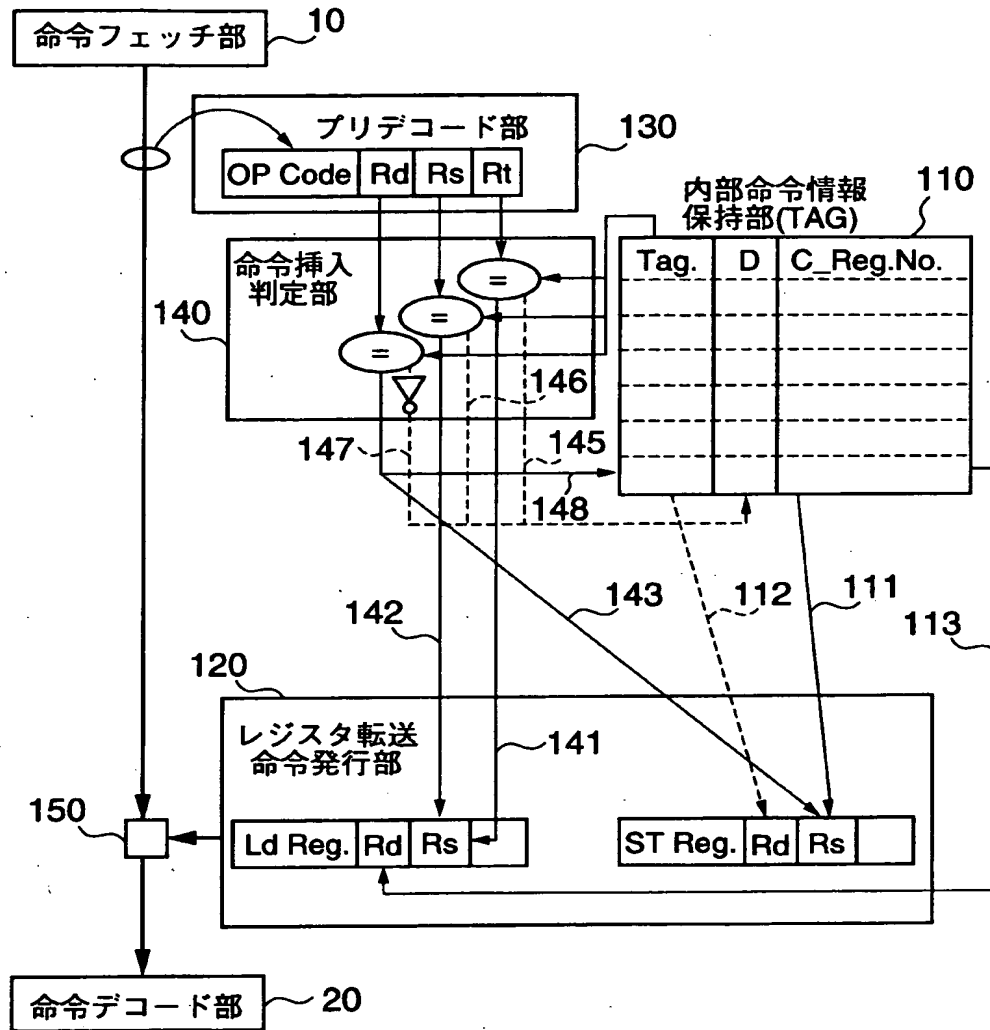
【図 5】



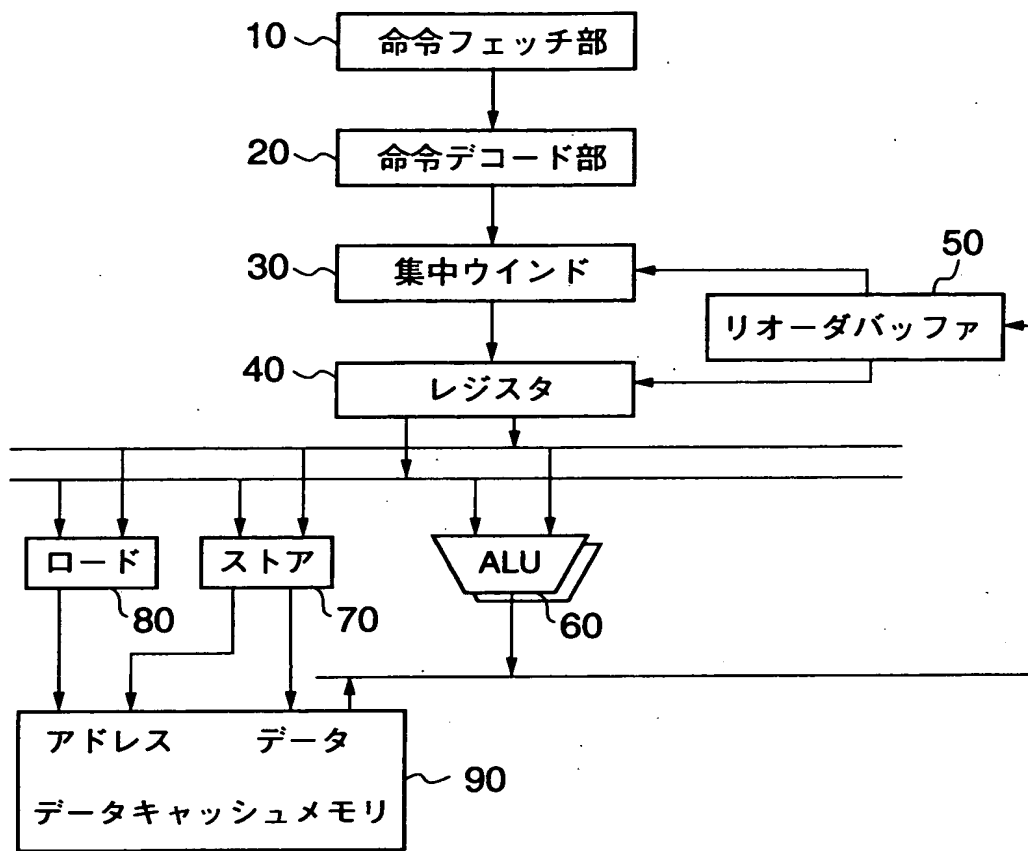
【図 6】



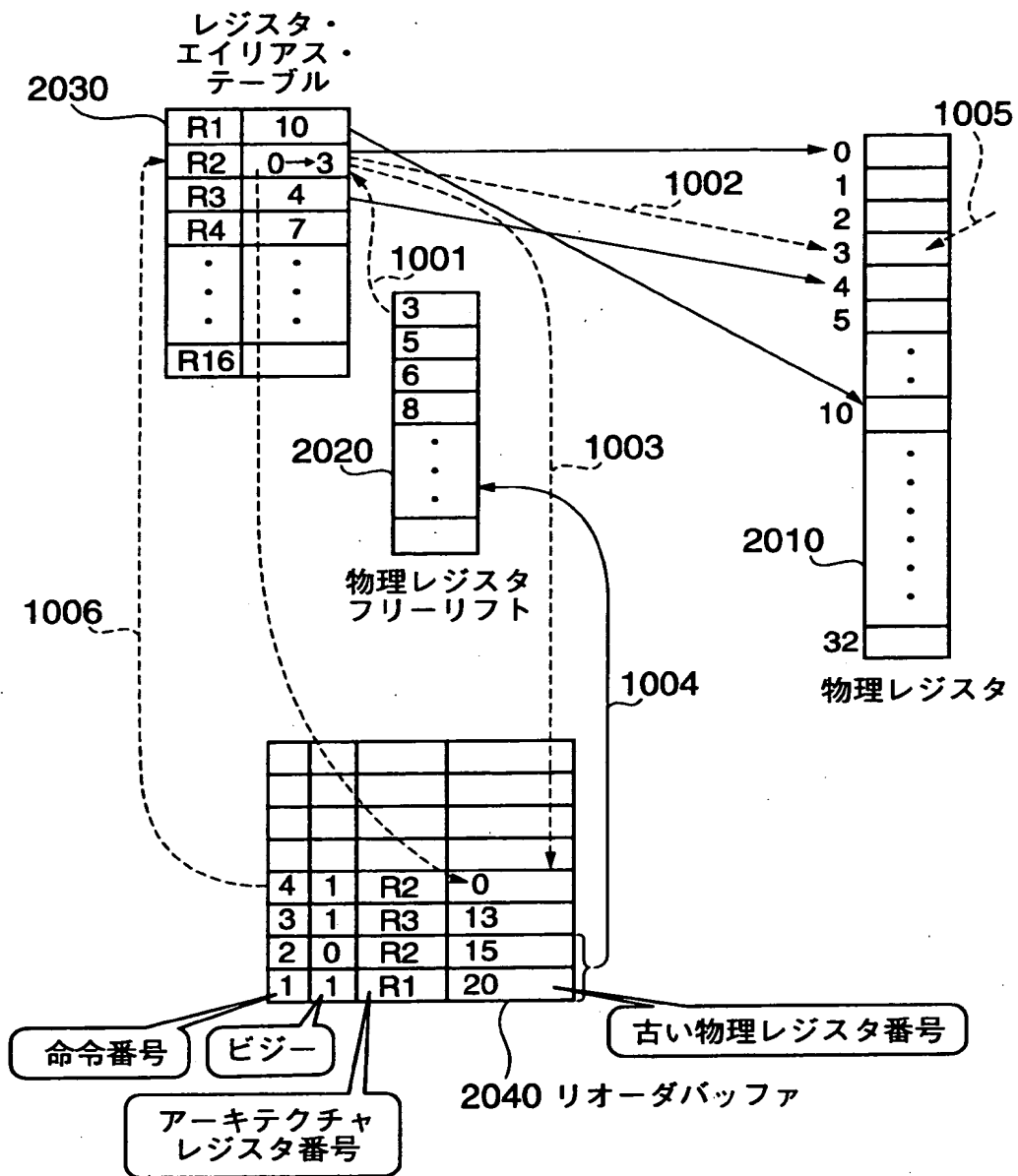
【図 7】



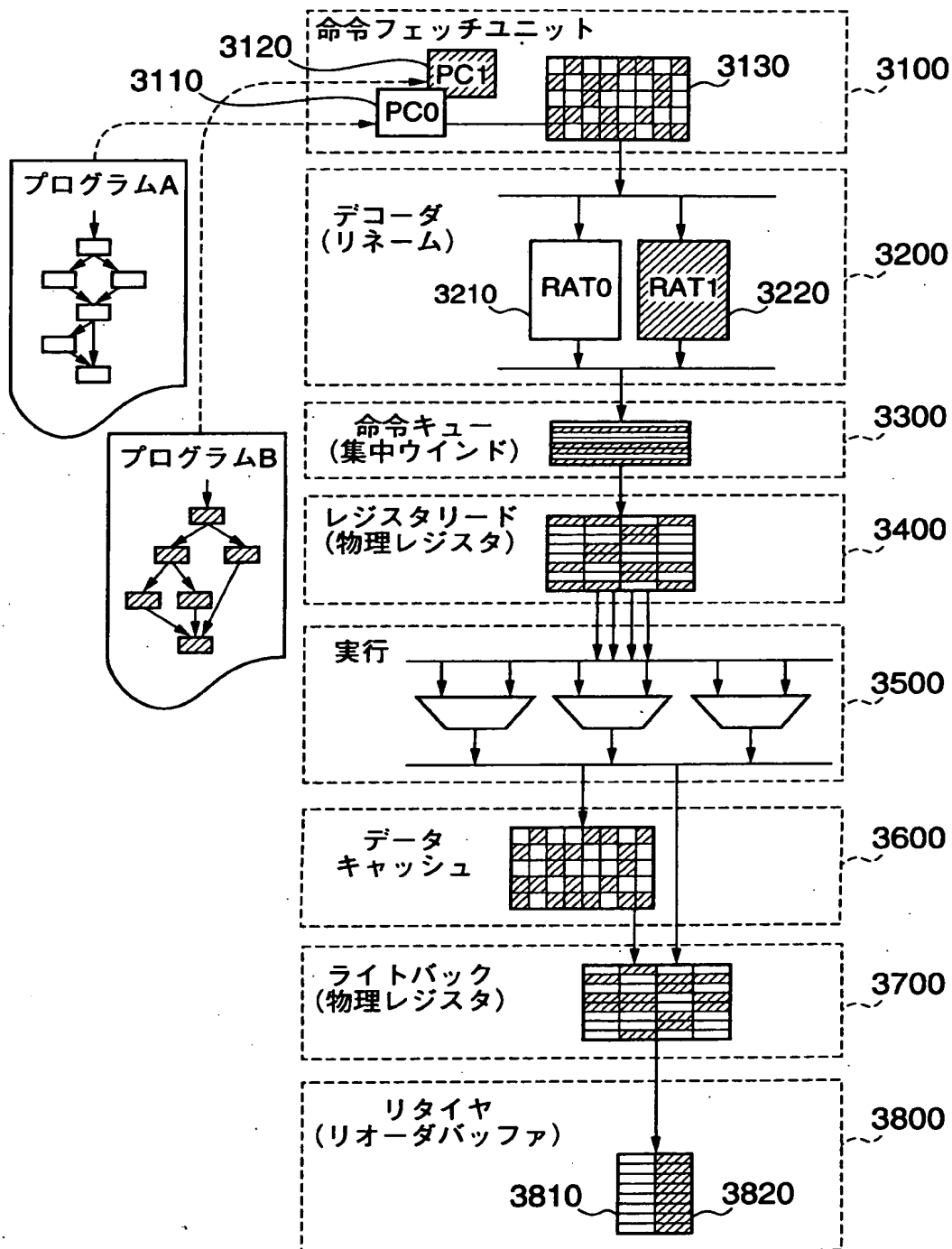
【図 8】



【図9】



【図10】



【書類名】 要約書

【要約】

【課題】 プログラム上で参照可能な論理レジスタ数を増加させても、アウト・オブ・オーダー (out-of-order) 実行等のような複雑な制御を可能にする。

【解決手段】 プロセッサは、命令フェッチ部 1 で取込まれた命令から命令コードを復号する命令デコード部 2 と、復号された命令コードが示すレジスタ番号のデータを保持するレジスタ本体 4 と、保持されたデータの一部をキャッシングするキャッシングレジスタ 3 と、フェッチ部 1 の命令によりキャッシングレジスタ 3 に保持された論理レジスタ番号やキャッシングレジスタ番号を含む内部命令状態の情報を保持する内部命令情報保持部 6 と、フェッチ部 1 からの命令をプリデコードした命令コードと保持部 6 で保持された内部命令の情報とを比較して内部命令を挿入するか否かを判定する命令挿入判定部 8 と、判定部 8 が内部転送命令の挿入を判定したときキャッシングレジスタ 3 とレジスタ本体 4 の間で内部データのレジスタ転送命令を発行するレジスタ転送命令発行部 9 とを備える。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000221199]

1. 変更年月日	1990年 8月23日
[変更理由]	新規登録
住 所	神奈川県川崎市川崎区駅前本町25番地1
氏 名	東芝マイクロエレクトロニクス株式会社

出 願 人 履 歴 情 報

識別番号 [000003078]

1. 変更年月日	2001年 7月 2日
[変更理由]	住所変更
住 所	東京都港区芝浦一丁目1番1号
氏 名	株式会社東芝